ซอฟต์แวร์
อนุชาติ ลาสนธิ
มุกดา สาริโส

SOFT MAIL

ถึงตัวโต... ก็เก่งนะ

สร้างตัวอักษรขนาดบิ๊ก เพื่อพิชิต..... ความเป็นหนึ่งของผลงาน

ท่านที่เคยเขียนโปรแกรมภาษาแอสเซมบลีคงจะเคยเรียกใช้ INT 10H ฟังก์ชันหมายเลข 09H มาแล้ว เป็นฟังก์ชันเขียนตัวอักษรหนึ่งตัวพร้อมแอตทริบิวต์บนจอภาพ ตัวอักษรที่ได้มีขนาด 9×14 สำหรับโมโนโครมหรือ 8×8 สำหรับจอสี แต่วันนี้ .....คุณจะได้รู้จัก INT 10H หมายเลขฟังก์ชัน 99H ที่คุณยังไม่เคยใช้มาก่อน เป็นฟังก์ชันที่ใช้สำหรับเขียนตัวอักษรหนึ่งตัวพร้อมแอตทริบิวต์ลงบนจอภาพเช่นกัน แต่ขนาดตัวอักษรที่ได้จะเป็น 7×12 ไบต์ เพื่อต้องการแสดงข้อความอะไรบางอย่างให้เห็นชัดเจน เช่น ข้อความใช้ เตือน ทักทาย หรือโชว์ข้อความสวย ๆ พร้อมลวดลายสักหนึ่งประโยคก่อนที่จะเข้าสู่โปรแกรม

ยุทธวิธีการใช้งาน

โปรแกรม BIG_CHAR.COM ในบทความนี้ เป็นโปรแกรมช่วยในการแสดงตัวอักษรขนาดใหญ่ทางจอภาพ เมื่อทำการติดตั้งหรือรันโปรแกรม BIG_CHAR.COM เครื่องของคุณจะมีโปรแกรมย่อยเพิ่มขึ้นสามารถใช้งานได้ โปรแกรมย่อยนี้จะเป็นหมายเลขฟังก์ชันที่ 99H ของ INT 10H การเรียกใช้ BIG_CHAR.COM ทำเหมือนกับการเรียกใช้อินเตอร์รัพต์หมายเลขอื่น ๆ ของไบออส คือ ต้องทราบข้อกำหนดในการเรียกใช้งาน สำหรับ BIG_CHAR.COM ต้องกำหนดค่าดังนี้

- รีจิสเตอร์ AH = 99H
- รีจิสเตอร์ AL = ค่าแอสกีของตัวอักษรที่จะพิมพ์
- รีจิสเตอร์ BL = แอตทริบิวต์
- รีจิสเตอร์ DL = ตำแหน่งคอลัมน์ที่จะพิมพ์
- รีจิสเตอร์ DH = ตำแหน่งแถวที่จะพิมพ์

ข้อควรระวัง

– รีจิสเตอร์ AL ต้องมีค่า 32-255 ถ้าค่าของรีจิสเตอร์ AL เป็น 0-31 ตัวอักษรที่พิมพ์จะเป็นตัวทึบ (แอสกี 219)

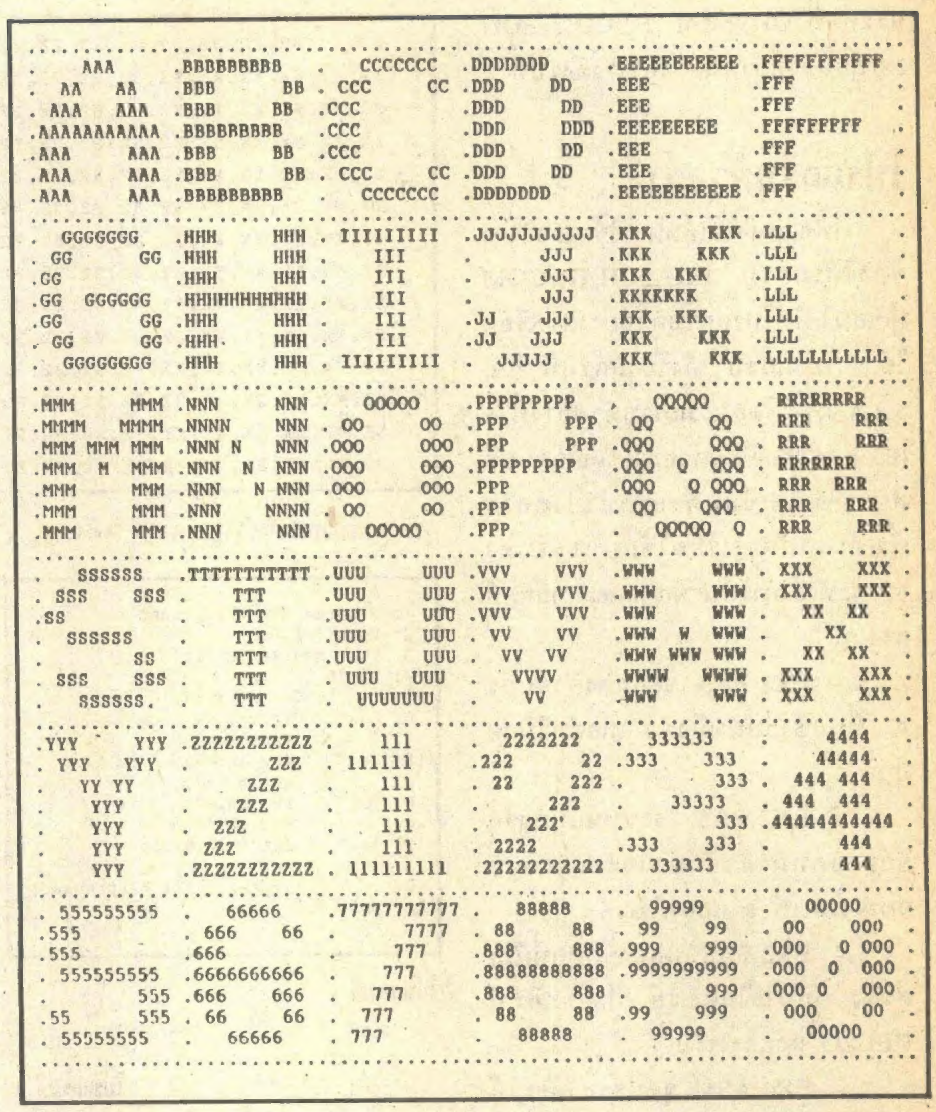
– ค่าของรีจิสเตอร์ DH หรือตำแหน่งแถวมีค่า 0-24

– ค่าของรีจิสเตอร์ DL หรือตำแหน่งคอลัมน์มีค่า 0-79

– ก่อนที่จะพิมพ์ตัวอักษร โปรแกรม BIG_CHAR.COM จะไม่มีการตรวจสอบค่าสองค่านี้ ดังนั้นในการเรียกใช้ให้ผู้ใช้ตรวจสอบเองว่าค่าเกินหรือยัง

FONT

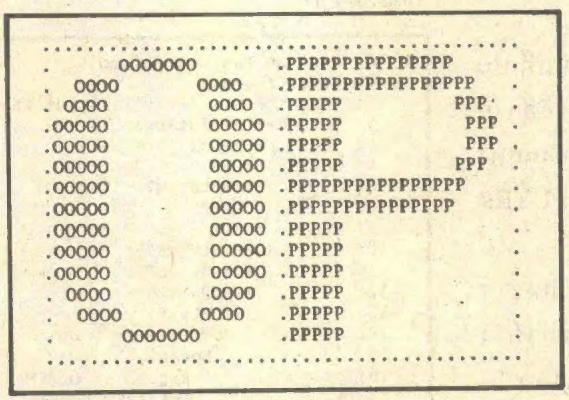
ฟอนต์ของ BIG_CHAR.COM นี้ออกแบบเป็น "ไบต์แม็พ" ขนาด 7×12 (row × col) ดังนั้นหนึ่งตัวอักษรจะมีขนาด 84 ไบต์ สามารถแสดงผลได้จากแอสกี 32-255 แต่โปรแกรมที่ส่งมานี้ออกแบบให้เพียง 37 ตัว คือ ช่องว่าง, A-Z, 0,9 ดังรูปที่ 1 ส่วนตัวอักษรที่เหลือให้ผู้ใช้ออกแบบกันเอาเองหรือจะแก้ฟอนต์ตัวเก่าที่ไม่ค่อยเหมือนก็ไม่ว่าอะไร ตัวที่ยังไม่ได้ออกแบบผู้เขียนกำหนดตามรหัสแอสกีไว้ให้เรียบร้อยแล้วและการออกแบบก็ต้องเรียงตามรหัสแอสกีด้วย ไม่เช่นนั้นโปรแกรมจะอ่านค่ามาแสดงผลผิดพลาด



รูปที่ 1 ฟอนต์ขนาด 7x12

วิธีเปลี่ยนขนาดของฟอนต์

ฟอนต์ที่ออกแบบไว้ 7×12 นี้ ถ้ายังไม่ถูกใจสามารถเปลี่ยนขนาดของฟอนต์ได้ตามต้องการ โดยจะต้องเปลี่ยนค่าบางอย่างในโปรแกรมบรรทัดที่ 19 F_ROW EQU 7 และบรรทัดที่ 20 F_COL EQU 12 ซึ่งสองค่านี้หมายถึงขนาดของฟอนต์ทางด้านแถวและคอลัมน์ตามลำดับนั่นเอง ดังตัวอย่างรูปที่ 2 ผู้เขียนออกแบบขนาดฟอนต์ใหม่เป็น 14×20 ดังนั้นโปรแกรมในบรรทัดที่ 19 จะเป็น F_ROW EQU 14 และ



รูปที่ 2 ฟอนต์ขนาด 14x20

บรรทัดที่ 20 จะเป็น F_COL EQU 20 ส่วนค่าที่กำหนดลงตารางดังรูปที่ 3

โปรแกรมตัวอย่าง

เพื่อแสดงให้เห็นการใช้งานจริง ๆ ของโปรแกรม BIG_CHAR.COM ผู้เขียนได้ให้โปรแกรมตัวอย่างการเรียกใช้ 7 โปรแกรม ดังโปรแกรมที่ 1,2, 3,4,5,6,7 โดยโปรแกรมตัวอย่างที่ 4 (EX4.ASM) เขียนด้วยภาษาแอสเซมบลี ส่วนที่เหลือเขียนด้วยภาษาปาสคาล เริ่มต้นจากการเรียกใช้แบบธรรมดา จนกระทั่งถึงการทำลวดลายและเอฟเฟกท์ต่าง ๆ

- EX1.PAS จะแสดงตัวอักษร A ที่มุมซ้ายบนหนึ่งตัว แอตทริบิวต์ปกติ

- EX2.PAS จะรับข้อมูลจากคีย์บอร์ดหนึ่งตัวอักษรและจะแสดงที่กลางจอภาพ 3 แอตทริบิวต์

- EX3.PAS แสดงข้อมูลที่เป็นสตริง เก็บไว้ในอะเรย์ โดยจะพิมพ์ HELLO และกะพริบ

- EX4.ASM จุดประสงค์จะให้เห็นตัวอย่างการเรียกใช้ด้วยภาษาแอสเซมบลี ซึ่งดูแล้วไม่กี่บรรทัดก็สามารถใช้ได้แล้ว

- EX5.PAS แสดงข้อมูลที่เป็นสตริงหลาย ๆ บรรทัด โดยบรรทัดแรกจะแสดง MICRO บรรทัดที่สองแสดง COM บรรทัดที่สามแสดง PUTER ทั้งสามมีแอตทริบิวต์ต่างกัน

โปรแกรมตัวอย่างที่ 6 และ 7 ที่เหลือ ต้องขอกล่าวถึงซอฟต์แวร์ตัวเก่าของผู้เขียนเองที่ลงในไมโครฯฉบับที่ 64 เรื่อง "เพิ่มฟังก์ชันม้วนจอให้ INT 10H" ซอฟต์แวร์ตัวดังกล่าวชื่อ ROLL.COM เป็นซอฟต์แวร์ประเภท

```
32,32,32,32,32,32,XX,XX,XX,XX,XX,XX,XX,32,32,32,32,32,32,32
32,32,XX,XX,XX,XX,32,32,32,32,32,32,32,XX,XX,XX,XX,32,32,32
32,XX,XX,XX,XX,32,32,32,32,32,32,32,32,XX,XX,XX,XX,32,32,32
XX,XX,XX,XX,XX,32,32,32,32,32,32,32,32,32,XX,XX,XX,XX,XX,32
XX,XX,XX,XX,XX,32,32,32,32,32,32,32,32,32,XX,XX,XX,XX,XX,32
XX,XX,XX,XX,XX,32,32,32,32,32,32,32,32,32,XX,XX,XX,XX,XX,32
XX,XX,XX,XX,XX,32,32,32,32,32,32,32,32,32,XX,XX,XX,XX,XX,32
XX,XX,XX,XX,XX,32,32,32,32,32,32,32,32,32,XX,XX,XX,XX,XX,32
XX,XX,XX,XX,XX,32,32,32,32,32,32,32,32,32,XX,XX,XX,XX,XX,32
XX,XX,XX,XX,XX,32,32,32,32,32,32,32,32,32,XX,XX,XX,XX,XX,32
XX,XX,XX,XX,XX,32,32,32,32,32,32,32,32,32,XX,XX,XX,XX,XX,32
32,XX,XX,XX,XX,32,32,32,32,32,32,32,32,32,XX,XX,XX,XX,32,32
32,32,XX,XX,XX,XX,32,32,32,32,32,32,32,XX,XX,XX,XX,32,32,32
32,32,32,32,32,32,XX,XX,XX,XX,XX,XX,XX,32,32,32,32,32,32,32
```

รูปที่ 3 ฟอนต์ของตัวอักษร 0 ขนาด 14x20

```
 1: Program Big_char_example1;
 2: Uses Crt,Dos;
 3: Var  Reg:Registers;
 4: Begin
 5:         TextAttr:=7;
 6:         Clrscr;
 7:          Reg.AH:=$99;
 8:          Reg.AL:=65;
 9:          Reg.BL:=7;
10:          Reg.DH:=0;
11:          Reg.DL:=0;
12:          Intr($10,Reg);
13:
14:         Repeat Until Keypressed;
15: End.
```

โปรแกรมที่ 1

```
 1: Program Big_char_example2;
 2: Uses Crt,Dos;
 3: Var  Reg:Registers;
 4:       Ch:Char;
 5: Begin
 6:         TextAttr:=7;
 7:         Clrscr;
 8:          Reg.AH:=$99;
 9:       Repeat
10:                Ch:=Readkey;
11:                Reg.AL:=Ord(Ch);
12:                Reg.BL:=7;
13:                Reg.DH:=10;
14:                Reg.DL:=10;
15:                Intr($10,Reg);
16:
17:                Reg.BL:=$78;
18:                Reg.DH:=10;
19:                Reg.DL:=30;
20:                Intr($10,Reg);
21:
22:                Reg.BL:=$FF;
23:                Reg.DH:=10;
24:                Reg.DL:=50;
25:                Intr($10,Reg);
26:
27:       Until Ch=' '
28:
29: End.
```

โปรแกรมที่ 2

โปรแกรมที่ 3

```
 1: Program Big_char_example3;
 2: Uses Crt,Dos;
 3: Const Mess : Array[1..5] of Char = ('H','E','L','L','O');
 4: Var  Reg:Registers;
 5:         I:Byte;
 6: Begin
 7:         TextAttr:=7;
 8:         Clrscr;
 9:
10:          Reg.AH:=$99;
11:          Reg.BL:=$FF;
12:          Reg.DH:=10;
13:          Reg.DL:=5;
14:            For i:=1 to 5 do
15:              Begin
16:                Reg.AL := Ord(Mess[i]);
17:                Intr($10,Reg);
18:                Reg.DL:=Reg.DL+13;
19:              End;
20:
21:         Repeat Until Keypressed;
22: End.
```

```
1: CSEG          SEGMENT
2:               ASSUME CS:CSEG,DS:CSEG
3:               ORG 100H
4: START:
5:
6:                MOV DH,10
7:                MOV DL,10
8:                MOV SI,OFFSET MESS
9:                MOV CX,5
10:               MOV BL,7
11:               MOV AH,99H
12: NEXT:
13:               MOV AL,[SI]
14:               INT 10H
15:               INC SI
16:               ADD DL,12
17:               LOOP NEXT
18:
19:               MOV AH,1
20:               INT 21H
21:
22:               MOV AH,4CH
23:               INT 21H
24:
25: MESS          DB 'SE ED'
26:
27: CSEG          ENDS
28:               END START
```

โปรแกรมที่ 4

โปรแกรมที่ 5

```
1: Program Big_char_example4;
2: Uses Crt,Dos;
3: Const Messl : Array[1..5] of Char = ('M','I','C','R','O');
4:       Mess2 : Array[1..3] of Char = ('C','O','M');
5:       Mess3 : Array[1..5] of Char = ('P','U','T','E','R');
6:
7: Var  Reg:Registers;
8:       I:Byte;
9: Begin
10:       TextAttr:=7;
11:       Clrscr;
12:
13:        Reg.AH:=$99; Reg.BL:=$FF; Reg.DH:=0; Reg.DL:=5;
14:          For i:=1 to 5 do
15:            Begin
16:              Reg.AL := Ord(Messl[i]);
17:              Intr($10,Reg);
18:              Reg.DL:=Reg.DL+13;
19:            End;
20:
21:        Reg.BL:=$78; Reg.DH:=9; Reg.DL:=15;
22:          For i:=1 to 3 do
23:            Begin
24:              Reg.AL := Ord(Mess2[i]);
25:              Intr($10,Reg);
26:              Reg.DL:=Reg.DL+13;
27:            End;
28:
29:        Reg.BL:=$7; Reg.DH:=18; Reg.DL:=5;
30:          For i:=1 to 5 do
31:            Begin
32:              Reg.AL := Ord(Mess3[i]);
33:              Intr($10,Reg);
34:              Reg.DL:=Reg.DL+13;
35:            End;
36:
37:        Repeat Until Keypressed;
38: End.
```

โปรแกรมที่ 6

```
1: Program Big_char_example6;
2: Uses Crt,Dos;
3: Const Mess : Array[1..5] of Char = ('H','E','L','L','O');
4: Var  Reg:Registers;
5:       I:Byte;
6: Begin
7:       TextAttr:=7;
8:       Clrscr;
9:
10:       Reg.AH:=$99; Reg.BL:=$7; Reg.DH:=10; Reg.DL:=5;
11:         For i:=1 to 5 do
12:           Begin
13:             Reg.AL := Ord(Mess[i]);
14:             Intr($10,Reg);
15:             Reg.DL:=Reg.DL+13;
16:           End;
17:         Delay(500);
18:
19:         Reg.AH:=$55; Reg.AL:=1; Reg.BH:=25;
20:         Reg.BL:=1;   Reg.CH:=1; Reg.CL:=1;
21:         Reg.DH:=25;  Reg.DL:=80;
22:         Intr($10,Reg);
23:
24:         Reg.AL:=1; Reg.BH:=25; Reg.BL:=2;
25:         Intr($10,Reg);
26:
27:         Reg.BH:=80;
28:         Reg.BL:=3;
29:         Intr($10,Reg);
30:
31:         Reg.BL:=4;
32:         Intr($10,Reg);
33:
34:       Repeat Until Keypressed;
35: End.
```

```
 1: Program Big_char_example7;
 2: Uses Crt,Dos;
 3: Const Mess1 : Array[1..4] of Char = ('L','E','F','T');
 4:       Mess2 : Array[1..5] of Char = ('R','I','G','H','T');
 5:       Mess3 : Array[1..2] of Char = ('U','P');
 6:       Mess4 : Array[1..4] of Char = ('D','O','W','N');
 7:
 8: Var    Reg:Registers;
 9:          I:byte;
10: Begin
11:       TextAttr:=7;
12:       Clrscr;
13:
14:        Reg.AH:=$99; Reg.BL:=$7; Reg.DH:=5; Reg.DL:=13;
15:          For i:=1 to 4 do
16:            Begin
17:              Reg.AL := Ord(Mess1[i]);
18:              Intr($10,Reg);
19:              Reg.DL:=Reg.DL+13;
20:            End;
21:
22:        Reg.DH:=15; Reg.DL:=5;
23:          For i:=1 to 5 do
24:            Begin
25:              Reg.AL := Ord(Mess2[i]);
26:              Intr($10,Reg);
27:              Reg.DL:=Reg.DL+13;
28:            End;
29:         Delay(500);
30:
31:         Reg.AH:=$55;  Reg.AL:=0;
32:           For i := 1 to 80*3 do
33:            begin
34:              Reg.CH:=1;  Reg.CL:=1;
35:              Reg.DH:=14; Reg.DL:=80;
36:              Reg.BH:=1;  Reg.BL:=3;
37:              Intr($10,Reg)
38:
39:              Reg.CH:=15; Reg.DH:=25;
40:              Reg.DL:=80; Reg.BL:=4;
41:              Intr($10,Reg);
42:            End;
43:
44:          Delay(1000);
45:       TextAttr:=7;
46:       Clrscr;
47:
48:        Reg.AH:=$99; Reg.BL:=$7; Reg.DH:=18; Reg.DL:=55;
49:          For i:=1 to 2 do
50:            Begin
51:              Reg.AL := Ord(Mess3[i]);
52:              Intr($10,Reg);
53:              Reg.DL:=Reg.DL+13;
54:            End;
55:
56:        Reg.AH:=$99; Reg.BL:=$7; Reg.DH:=0; Reg.DL:=0;
57:          For i:=1 to 4 do
58:            Begin
59:              Reg.AL := Ord(Mess4[i]);
60:              Intr($10,Reg);
61:              Reg.DL:=Reg.DL+13;
62:            End;
63:
64:         Delay(500);
65:         Reg.AH:=$55;  Reg.AL:=1;
66:           For i := 1 to 50 do
67:            begin
68:              Reg.CH:=1;  Reg.CL:=1;
69:              Reg.DH:=25; Reg.DL:=54;
70:              Reg.BH:=1;  Reg.BL:=1;
71:              Intr($10,Reg);
72:
73:              Reg.CH:=1;  Reg.CL:=55;
74:              Reg.DH:=25; Reg.DL:=80;
75:              Reg.BL:=2;
76:              Intr($10,Reg);
77:            End;
78:
79:       Repeat Until Keypressed;
80: End.
```

โปรแกรมที่ 7

เดียวกันกับ BIG_CHAR.COM เมื่อติดตั้งแล้วจะเป็นหมายเลขฟังก์ชัน 55H ของ INT 10H เช่นกัน เมื่อจะเรียกใช้ต้องกำหนดรีจิสเตอร์ดังนี้

- รีจิสเตอร์ AH=55H
- รีจิสเตอร์ AL เวลาหน่วง
- รีจิสเตอร์ BH จำนวนครั้งที่จะม้วนจอ
- รีจิสเตอร์ BL รูปแบบของการม้วน
- รีจิสเตอร์ CH ค่าของ Y1
- รีจิสเตอร์ CL ค่าของ X1
- รีจิสเตอร์ CH ค่าของ Y2
- รีจิสเตอร์ DL ค่าของ X2
- EX6.PAS จะแสดง HELLO ที่กลางจอ สักครู่จะเลื่อนลงขึ้นซ้ายและขวาอย่างละหนึ่งรอบ โดยการเลื่อนตัวอักษรนี้ใช้เทคนิคการม้วนจอนั่นเอง
- EX7.PAS โปรแกรมตัวอย่างสุดท้าย ถึงแม้จะยาวสักหน่อยแต่ผลการรันก็น่าดูชมไม่เบา ลองรันโปรแกรมดูนะครับ

Source ไฟล์

BIG_CHAR.COM เขียนด้วยภาษาแอสเซมบลีตระกูล 8086 ซอร์สไฟล์ที่เป็นนามสกุล .ASM นี้จะยาวประมาณห้าร้อยกว่าบรรทัด แต่ไม่ใช่เป็นโปรแกรมทั้งหมดนะครับสามร้อยกว่าบรรทัดแรกเป็นรูปร่างของฟอนต์ โปรแกรมจริง ๆ มีเพียงร้อยบรรทัดเท่านั้นโดยแบ่งเป็น 7 ส่วนดังนี้

- BIG_CHAR PROC FAR เป็นโปรแกรมหลักทำหน้าที่ตรวจสอบ

ว่ามีการใช้ INT 10H หรือไม่ ถ้าใช่ก็จะตรวจสอบค่าของรีจิสเตอร์ AH ถ้ามีค่าเป็น 99H จะเริ่มทำงานตามโปรแกรม คือ เก็บค่าในรีจิสเตอร์ DL (col) DH (row) BL (attribute) AL (char) เก็บตำแหน่งเคอร์เซอร์ในขณะนั้นไว้ คำนวณหาค่าเพื่อให้รีจิสเตอร์ SI ไปชี้ที่ตารางฟอนต์ของตัวอักษรในรีจิสเตอร์ AL แล้วพิมพ์ตัวอักษรขนาดใหญ่ลงบนจอภาพในตำแหน่ง DH และ DL ก่อนออกจากโปรแกรม จะคืนตำแหน่งเคอร์เซอร์ให้เหมือนเดิม

- SAVE_POS PROC NEAR เก็บตำแหน่งเคอร์เซอร์ขณะนั้นไว้ในตัวแปร POSITION

- ADJUSE PROC NEAR คำนวณค่าในรีจิสเตอร์ AL เพื่อให้รีจิสเตอร์ SI ชี้ตาราง

- SET_POS PROC NEAR เซตเคอร์เซอร์ให้อยู่ในตำแหน่ง Row, Col

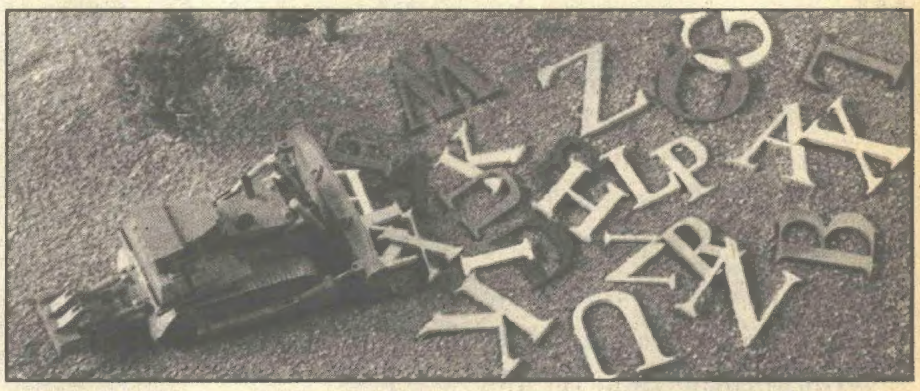
- PRINT PROC NEAR แสดงตัวอักษรธรรมดาหนึ่งตัวพร้อมแอตทริบิวต์ที่ตำแหน่งเคอร์เซอร์

- SHOW PROC NEAR อ่านค่าในตารางมาแสดงเพื่อทำตัวอักษรขนาดใหญ่

- INSTALL ส่วนนี้จะทำหน้าที่ฝังโปรแกรมและรูปร่างของฟอนต์ทั้งหมดในหน่วยความจำก่อนที่จะออกจากโปรแกรมสู่ดอส

```
C:\ANUCHAT\ASM\DEV>MASM BIG_CHAR;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.


  51556 + 324796 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\ANUCHAT\ASM\DEV>LINK BIG_CHAR;

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

LINK : warning L4021: no stack segment

C:\ANUCHAT\ASM\DEV>EXE2BIN BIG_CHAR.EXE BIG_CHAR.COM
```

รูปที่ 4 การแอสเซมเบลอร์ด้วย MASM VER 5.0



คอมพายล์โปรแกรม

โปรแกรมแอสเซมเบลอร์ผู้เขียนใช้ MASM รุ่น 5.0 ผลการแอสเซมเบลอร์ดังรูปที่ 4 หรือถ้าจะใช้เทอร์โบแอสเซมเบลอร์ก็คงไม่มีปัญหาอะไร

บทความนี้ได้อ้างถึงซอฟต์แวร์ตัวเก่า ดังนั้นถ้าจะทดลองรันโปรแกรม EX6, EX7 จะต้องจัดหาโปรแกรม ROLL.COM ก่อน และรันพร้อมกันทั้งสองโปรแกรม คือ BIG_CHAR.COM และ ROLL.COM โปรแกรมทั้งหมดนี้ทางกองบรรณาธิการได้จัดทำเป็นซอฟต์เมล์บริการท่านผู้อ่านแล้วครับ!